



Öğr.Gör. Mustafa GÖKMEN
gokmen@selcuk.edu.tr



Nesneye Yönelik Programlama

Arrays





Diziler (Arrays)

- ✓ Array, **aynı tipten çok sayıda değişken** tanımlamak için kullanılır. Java dilinde **Arrays** bir sınıftır (*java.util* paketinde). Arrays sınıfı **dizi oluşturma**, dizilerle **işlem yapma**, dizi içinde **bileşen arama** ve dizinin **bileşenlerini sıralama** gibi dizi ile ilgili işlemleri yapmaya yarayan öğeleri içeren bir sınıftır.
- ✓ Dizi elemanlarına aynı isim üzerinden **indis** kullanılarak **erişilebilir** çünkü bu elemanların **bellekte ardışık** olarak tutulmaktadır. Örneğin 100 elemanlı bir tamsayı (**int**) dizisi oluşturulduğunda dizi elemanlarına 400 *Byte* uzunluğunda bir bellek bloğu ayrılmış demektir.
- ✓ Bir diziyi gösteren değişkenin tanımı şu şekilde yapılır.

```
Eleman_Tipi[] Dizi_Adi;
```

veya

```
Eleman_Tipi Dizi_Adi[];
```



Dizi Elemanlarına Değer Atama Yöntemleri

1: Array'in uzunluğunu belirleyip index ile değer atama

```
int[] arr;           // 1 ve 2. satırlar tek seferde  
arr = new int[10];   // int[] arr = new int[10]; olabilir  
arr[6]=70; arr[1]=20; arr[2]=30; arr[7]=80; arr[4]=50;  
arr[3]=40; arr[0]=10; arr[8]=90; arr[9]=60; arr[5]=100;
```

2: Nesneyi doğrudan oluşturup değer atama

```
int[] arr = new int[] { 1234, 2746, 1785, 8732, 1079 }; //Explicit
```

3: Nesneyi dolaylı oluşturup değer atama

```
int[] arr = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 }; //Implicit
```



Örnek: sort() metodu ile sıralama

```
import java.util.Arrays;

public class ArraySort {
    public static void diziYaz(double[] dbl) {
        for (int i = 0; i < dbl.length; i++) {
            System.out.println("dbl[" + i + "] = " + dbl[i]);
        }
    }

    public static void main(String[] args) {
        double[] dizi = { 2, 3.45, -4.78, 1, 15.12, -15.32, 78.17 };
        System.out.println("Karışık : ");
        diziYaz(dizi);
        Arrays.sort(dizi);
        System.out.println("Sıralı:");
        diziYaz(dizi);
    }
}
```

Karışık :

dbl[0] = 2.0
dbl[1] = 3.45
dbl[2] = -4.78
dbl[3] = 1.0
dbl[4] = 15.12
dbl[5] = -15.32
dbl[6] = 78.17


Sıralı:

dbl[0] = -15.32
dbl[1] = -4.78
dbl[2] = 1.0
dbl[3] = 2.0
dbl[4] = 3.45
dbl[5] = 15.12
dbl[6] = 78.17



Örnek: Dizi kopyalama (geleneksel)

```
public class ArrayCopyIndex {  
    public static void main(String[] args) {  
        int[] arr1 = { 0, 2, 4, 6 };  
        int[] arr2 = new int[arr1.length];  
        for (int i = 0; i < arr1.length; i++) {  
            arr2[i] = arr1[i];  
        }  
  
        System.out.println("-- Kopya --");  
  
        for (int i = 0; i < arr2.length; i++) {  
            System.out.printf("arr2[%d]=%d\n", i, arr2[i]);  
        }  
    }  
}
```



```
-- Kopya --  
arr2[0]=0  
arr2[1]=2  
arr2[2]=4  
arr2[3]=6
```



Örnek: Dizi kopyalama (arraycopy())

```
public class ArrayCopyMethod {  
    public static void main(String[] args) {  
        int[] arr1 = { 1, 2, 3, 4 };  
        int[] arr2 = { 80, 70, 60, 50, 40, 30, 20, 10 };  
        System.arraycopy(arr1, 2, arr2, 2, arr1.length - 2);  
  
        for (int i = 0; i < arr2.length; i++) {  
            System.out.print(arr2[i] + " ");  
        }  
    }  
}
```

80 70 3 4 40 30 20 10

```
arraycopy(srcArray, srcStartPos, dstArray, dstStartPos, length);  
  
arr1 = { 1, 2, 3, 4 };  
  
arr2 = { 80, 70, 60, 50, 40, 30, 20, 10 };  
  
arr2 = { 80, 70, 3, 4, 40, 30, 20, 10 };
```





Örnek: Dizi klonlama (clone() ile)

```
public class ArrayClone {  
    public static void main(String[] args) {  
        int[] arrOrig = new int[] { 1, 3, 5 };  
        int[] arrClone = arrOrig.clone();  
        System.out.print("Diziler aynı");  
  
        if (arrOrig != arrClone) {  
            System.out.print(" değil");  
        }  
  
        System.out.println("dir.");  
  
        for (int i = 0; i < arrOrig.length; i++) {  
            System.out.println("arrOrig[" + i + "] = " + arrOrig[i] + " ");  
        }  
  
        for (int i = 0; i < arrClone.length; i++) {  
            System.out.println("arrClone[" + i + "] = " + arrClone[i] + " ");  
        }  
    }  
}
```

Diziler aynı değildir.

arrOrig[0] = 1

arrOrig[1] = 3

arrOrig[2] = 5

arrClone[0] = 1

arrClone[1] = 3

arrClone[2] = 5



Çok Boyutlu Diziler

Diziler birden fazla boyutlu olabilir. Örneğin bir matrisi ifade etmek üzere iki boyutlu bir dizi tanımlanmalıdır.

```
int[][] matrix;  
matrix = new int[5][10];
```

Bu tanımlama, 5 satırı olan, her satırında 10 sütun bulunan bir matris oluşturmaktadır ve şu şekilde de yazılabilir:

```
int[][] matrix = new int[5][10];
```

Matrisin elemanlarına erişim için iki indis kullanılır (indisler her boyut için 0 (sıfır)'dan başlar. İlk indis satır numarasını, ikinci indis sütun numarasını ifade eder. Aşağıdaki kod kesimi matrisin ilk satırının ilk sütununa 3, son satırının son sütununa ise 5 değerini aktarmaktadır.

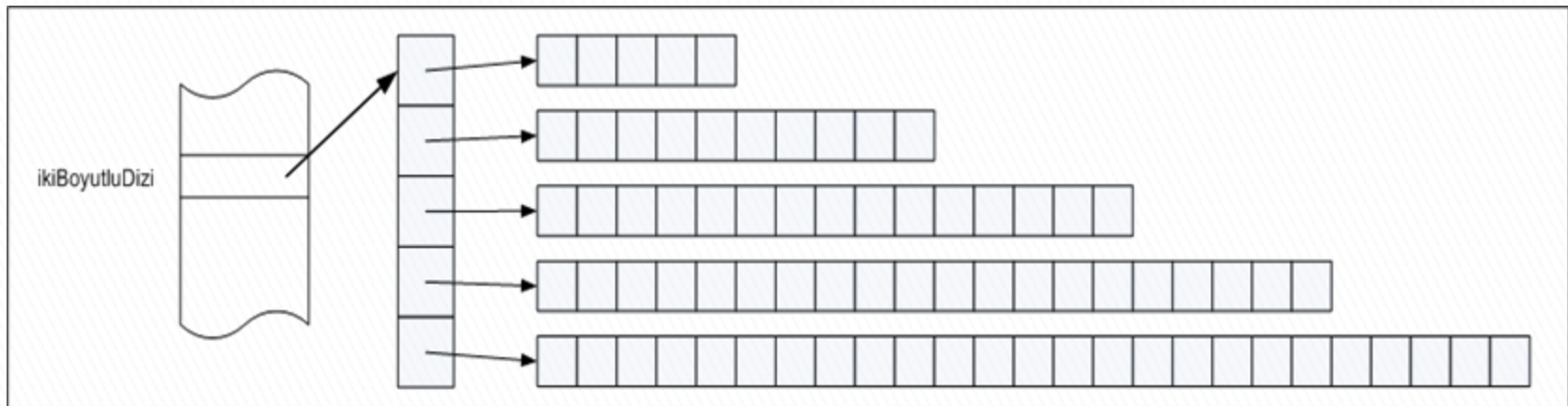
```
matrix[0][0] = 3;  
matrix[4][9] = 5;
```




Çok Boyutlu Diziler

Çok boyutlu dizilerin her boyutu, ayrı ayrı da oluşturulabilir. Bu durumda, her satırında farklı sayıda sütun olan iki boyutlu bir dizi şu şekilde oluşturulabilir.

```
int[][] ikiBoyutluDizi = new int[5][];  
ikiBoyutluDizi[0] = new int[5];  
ikiBoyutluDizi[1] = new int[10];  
ikiBoyutluDizi[2] = new int[15];  
ikiBoyutluDizi[3] = new int[20];  
ikiBoyutluDizi[4] = new int[25];
```





Örnek: Çok Boyutlu Diziler

```
public class ArrayMulDim {  
    public static void main(String args[]) {  
        int[][] arr = new int[3][4];  
        arr[0][0] = 45; arr[0][1] = 83; arr[0][2] = 11; arr[0][3] = 18;  
        arr[1][0] = 17; arr[1][1] = 56; arr[1][2] = 26; arr[1][3] = 79;  
        arr[2][0] = 13; arr[2][1] = 93; arr[2][2] = 43; arr[2][3] = 12;  
  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = 0; j < arr[i].length; j++) {  
                System.out.printf("%d    ", arr[i][j]);  
            }  
            System.out.println();  
        }  
    }  
}
```

45	83	11	18
17	56	26	79
13	93	43	12






Örnek: Çok Boyutlu Diziler

Aşağıdaki programda, iki boyutlu bir dizi oluşturulmaktadır. İlk boyutu 3 elemanlıdır. İkinci boyutundaki eleman sayıları ise farklıdır. Aşağıdaki örnekte oluşturulan dizi, 3 satırlı ve sırasıyla 3, 2 ve 4 sütunludur.

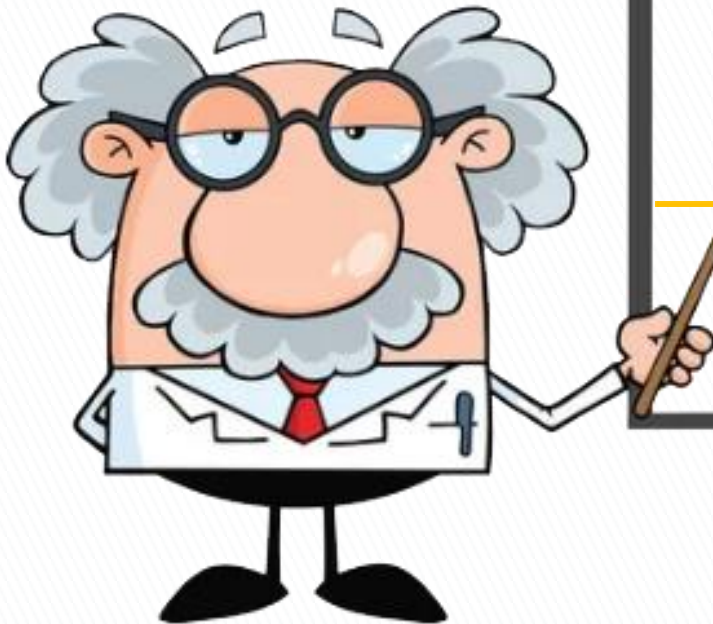
```
public class ArrayMulDiff {  
    public static void main(String[] args) {  
        int[][] arr = {  
            { 1, 2, 3 },  
            { 4, 5 },  
            { 6, 7, 8, 9 }  
        };  
  
        for (int i = 0; i < arr.length; i++) {  
            for (int j = 0; j < arr[i].length; j++) {  
                System.out.printf("%d ", arr[i][j]);  
            }  
            System.out.println();  
        }  
    }  
}
```



1	2	3	
4	5		
6	7	8	9



Öğr.Gör. Mustafa GÖKMEN
gokmen@selcuk.edu.tr



Nesneye Yönelik Programlama

Randomization





Rastgele Sayı Üretmek

Java'da rastgele sayı üretmek için *Random* sınıfının içinde *next* metodları kullanılır. Scanner sınıfında olduğu gibi *java.util.Random* sınıfı import edilmelidir.

```
Random rnd = new Random();
```

ile oluşturulan *rnd* referans nesnesi ile işlemler gerçekleştirilir.

```
...  
Random rnd = new Random();  
...  
int[] dizi1 = new int[10];  
dizi1[i] = rnd.nextInt(100);  
...  
double[] dizi2 = new double[10];  
dizi2[i] = rnd.nextDouble();  
...
```



Rastgele Sayı Üretmek

Rastgele bir sayı üretmenin bir diğer yolu da `java.lang` paketindeki `Math` sınıfında bulunan `random()` metodudur. Ancak bu metod `0` ile `1` arasında ondalıklı bir değer üretir. Bu metod static olup sınıf adı ile kullanılır ve `java.lang.Math` sınıfı import edilmesi gerekmez.

```
import java.util.Random;

public class ArrayRandomize {
    public static void main(String[] args) {
        Random rnd = new Random();
        int[] myArray = new int[10];
        for (int i = 0; i < myArray.length; i++) {
            myArray[i] = rnd.nextInt(10000);
            System.out.println(myArray[i]);
        }
        System.out.println(Math.random());
        System.out.println(Math.random() * 10);
        System.out.println((int) Math.random()); // 0 ?
        System.out.println((int) (Math.random() * 10));
    }
}
```

```
1049
1574
1774
9223
581
7265
6188
6702
8469
8512
0.8635458599893319
4.509809449115578
0
6
```

```
3769
5301
9815
7307
9299
2727
6163
2102
565
5472
0.7305170241258904
1.2739086364224916
0
8
```





Öğr.Gör. Mustafa GÖKMEN
gokmen@selcuk.edu.tr



Nesneye Yönelik Programlama


`main()` parameters





Örnek: Java main() metodu

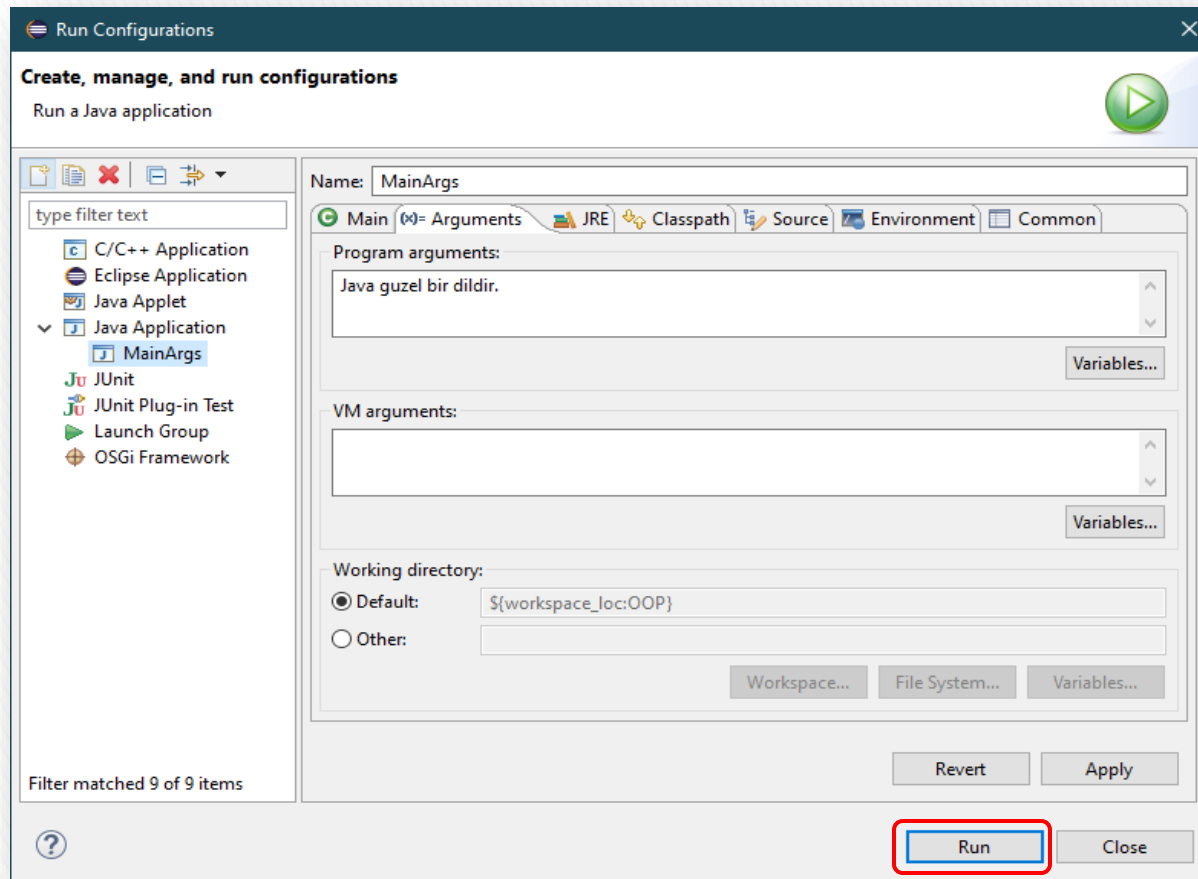
```
public class MainArgs {  
    public static void main(String[] args) {  
        if (args.length == 0) {  
            System.err.println("HATA: Parametre girmediniz!..");  
            return; // Attention - A void method  
        }  
  
        System.out.print("Parametre Sayısı: " + args.length + "\n");  
  
        int a = 0;  
  
        while (a < args.length) {  
            System.out.println("args[" + a + "] = " + args[a++]);  
        }  
    }  
}
```



Parametre Sayısı: 4
args[0] = Java
args[1] = guzel
args[2] = bir
args[3] = dildir.



Örnek: Java main() metodu



Not: Programın doğru bir şekilde çalışabilmesi için menüden **Run→Run Configurations→Arguments** komutu parametreler eklenmelidir. Daha sonra **Run** butonu ile çalıştırılabilir.



Örnek: Java main() metodu

The image shows a Windows command prompt window titled "Administrator: C:\Windows\system32\cmd.exe" and a File Explorer window titled "ders05".

In the command prompt, the directory path is shown as:

```
tr
  edu
    selcuk
      ceng
        ders05
```

The command executed is:

```
C:\JAVA>java tr.edu.selcuk.ceng.ders05.MainArgs Java guzel bir dildir.
```

The output shows the arguments passed to the program:

```
Parametre Sayisi: 4
args[0] = Java
args[1] = guzel
args[2] = bir
args[3] = dildir.
```

The File Explorer window shows the directory structure:

```
tr > edu > selcuk > ceng > ders05
```

The file `MainArgs.class` is listed, which is a Java Class File (1,01 KB).

A text box at the bottom of the File Explorer window contains the Java command syntax:

```
java <paket>.<sinif> [arg0 ... argN]
```